

特表2004-515001

(P2004-515001A)

(43) 公表日 平成16年5月20日 (2004.5.20)

(51) Int. Cl. 7	F I	テーマコード (参考)
G06F 12/14	G06F 12/14 320B	5B017
H03K 19/173	H03K 19/173 101	5J042
H04L 9/10	H04L 9/00 621A	5J104

審査請求 未請求 予備審査請求 有 (全 90 頁)

(21) 出願番号 特願2002-546876 (P2002-546876)  
 (86) (22) 出願日 平成13年11月28日 (2001.11.28)  
 (85) 翻訳文提出日 平成15年5月15日 (2003.5.15)  
 (86) 国際出願番号 PCT/US2001/045056  
 (87) 国際公開番号 W02002/044876  
 (87) 国際公開日 平成14年6月6日 (2002.6.6)  
 (31) 優先権主張番号 09/724,652  
 (32) 優先日 平成12年11月28日 (2000.11.28)  
 (33) 優先権主張国 米国 (US)  
 (81) 指定国 EP (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), CA, JP

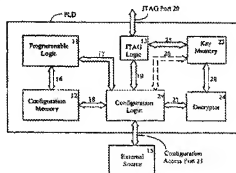
(71) 出願人 591025439  
 サイリンクス インコーポレイテッド  
 X I L I N X I N C O R P O R A T E D  
 アメリカ合衆国 カリフォルニア州 95  
 124-3400 サン ホセ ロジック  
 ドライブ 2100  
 (74) 代理人 100064746  
 弁理士 深見 久郎  
 100085132  
 (74) 代理人 弁理士 森田 俊雄  
 100083703  
 (74) 代理人 弁理士 仲村 義平  
 100096781  
 (74) 代理人 弁理士 堀井 豊

最終頁に続く

(54) 【発明の名称】 解読アルゴリズムおよび解読鍵を有するプログラマブルロジックデバイス

## (57) 【要約】

プログラマブルロジックデバイス (PLD) に実現される設計の複製を防ぐために、PLD 自体が、設計者がロードした解読鍵を記憶し、暗号化された構成ビットストリームが PLD にロードされる際にこれを解読するための解読器を含む。PLD はまた、ビットストリームが暗号化されているかどうかを示すヘッダ情報を脱出するためのロジックを含み、暗号化されたビットストリームも暗号化されないビットストリームも受け取ることができる。暗号化鍵は、不揮発性メモリに記憶されてもよく、または電源が抜かれたときに保持されるように電池でバックアップされてもよい。



**【特許請求の範囲】****【請求項1】**

暗号化されたビットストリームを解読するための解読器と、解読器によって用いられる鍵とを有するPLDにおいて、PLDを用いる方法であって、PLDを非安全モードにするステップと、鍵をPLDにロードするステップとを含む、方法。

**【請求項2】**

鍵をロードするステップの後にPLDを安全モードにするステップをさらに含む、請求項1に記載のPLDを使用する方法。

**【請求項3】**

構成メモリにより構成される構成可能ロジックと、PLDの外部源からビットストリームを受けるための構造と、解読鍵を記憶するための鍵メモリと、鍵を用いてビットストリーム中の暗号化された構成ビットを解読することにより構成データを形成するための解読アルゴリズムを有する解読器と、構成データを構成メモリにロードするための構造とを含む、プログラマブルロジックデバイス（PLD）。

**【請求項4】**

構成データを構成メモリにロードするための構造は、CRCチェックサム計算回路を含む、請求項3に記載のPLD。

**【請求項5】**

構成メモリから構成をリードバックするための構造と、ビットストリームが暗号化されたデータを含むことをヘッダ情報が示すとき構成をリードバックするための構造を不能化するための構造とをさらに含む、請求項3に記載のPLD。

**【請求項6】**

PLDが構成された後PLDを再構成するための構造と、ビットストリームが暗号化されたデータを含むことをヘッダ情報が示すときPLDを再構成するための構造を不能化するための構造とをさらに含む、請求項3に

記載の P L D。

【請求項 7】

解読器は、複数個の解読鍵を記憶するためのレジスタの 1 つから、解読のために別の鍵も使用されるかどうかを示す値を讀出す、請求項 3 に記載の P L D。

【請求項 8】

解読器は、鍵によって特定される態様と異なったように鍵を用いようとするならば、解読を中止するための回路を含む、請求項 3 に記載の P L D。

【請求項 9】

鍵は、鍵の組のうち最初の鍵か、中間の鍵か、最後の鍵か、または唯一の鍵かを特定する、請求項 3 に記載の P L D。

## 【発明の詳細な説明】

## 【0001】

## 〔発明の分野〕

この発明はPLDに関し、より特定的にはビットストリームによりPLDにロードされた設計の保護に関する。

## 【0002】

## 〔発明の背景〕

PLD（プログラマブルロジックデバイス）は、設計者によって選択されるデジタル論理機能を行なう集積回路構造である。PLDは、論理ブロックおよび配線を含み、典型的には論理ブロックも配線もプログラム可能である。PLDの一般的なタイプの1つはFPGA（フィールドプログラマブルロジックデバイス）で、論理ブロックが、典型的にはルックアップテーブルおよびフリップフロップを含み、典型的にはそれらの入力信号のいかなる機能も生成し記憶することができる。別のタイプはCPLD（コンプレックスプログラマブルロジックデバイス）で、論理ブロックがAND機能およびOR機能を行ない、入力信号の選択がプログラム可能である。

## 【0003】

PLDの外部のビットストリームを記憶することの問題

PLDにおいて実現される設計は複雑になってきており、設計を完成しデバックしPLD内で実現するのに何ヶ月もかかることがしばしばある。設計が、PLDが一部であるシステムとなり、利潤を得るために販売される場合、設計者はこの設計努力の結果が他人に複製されることを望まない。設計者はその設計をトレードシークレットにすることをしばしば望む。多くのPLD、特にFPGAは、PLDに電源が投入されるたびに、PROMなどの外部装置からロードされなければならない揮発性構成メモリを使用する。構成データはPLDの外部に記憶され構成アクセスポートを介して送信されなければならないので、たとえば基板パターン上にプローブを置くことで、構成アクセスポート上のデータをモニタする攻撃者により設計のプライバシーはたやすく侵害されかねない。

## 【0004】

### 現在の解決策およびその欠点

設計を暗号化する努力がなされてきたが、設計を攻撃者から安全にしかつ合法的ユーザによって使用しやすくすることは困難である。暗号化アルゴリズムは問題ではない。いくつかの暗号化アルゴリズム、たとえば、標準データ暗号化規格（DES）およびより安全な高度暗号化規格（AES）アルゴリズムが、データのブロックを暗号化するものとして公知である。暗号化されていないデータワードと次の暗号化されたデータワードとの排他的論理和を取りその後に解読をする暗号ブロック連鎖（CBC）のプロセスにより、DESまたはAESは、データの直列ストリームを暗号化することができ、したがってこれらはビットストリームを暗号化してPLDを構成するのに適切である。設計を暗号化するために使用される鍵は、PLDと設計を解読する構造との間で安全な態様で何らかの形でやり取りされなければならないので、その設計は、PLDを構成するために使用される前にPLDによって解読され得る。そして、暗号化されていない設計を用いて一旦PLDが構成されてしまうと、設計は引続き無許可の発見から保護されなければならない。

#### 【0005】

ザイリンクス・インコーポレイテッド（Xilinx, Inc.）のPeter Alfkeによる、「構成の問題：電源投入、揮発性、安全性、電池バックアップ」（“Configuration Issues: Power-up, Volatility, Security, Battery Back-up”）と題する1997年11月24日付の刊行物は、FPGA内に設計を保護するための特定のアーキテクチャの特徴を持たない既存のFPGAデバイス内の設計を保護するために取り得るいくつかのステップを記載する。1つの方法は、設計構成データをFPGAにロードし、次に構成データのソースを除去するが、電池を用いてFPGAへ連続的な電源を維持しながらFPGAを待機非動作モードに保持することである。しかしながら、電池に対する電源要求のため、この方法は大規模なFPGAデバイスには実践的でない。

#### 【0006】

別の可能性は、不揮発性構成メモリである。デバイスを販売する前に設計を工場

でロードする場合、構成されたPLDデバイスの購入者が、設計が何であるかを判断することは難しい。しかしながら、リバーシエンジニアリング工程により、すなわち、プログラムされたデバイスの覆いを取除き、金属層を除去し、不揮発性メモリセルを化学的に処理することで、どのメモリセルが充電されているのが暴露され、そのため攻撃者は設計を知ることができてしまう。さらに、不揮発性メモリは、標準のCMOSプロセス技術よりもより複雑でより高価なプロセス技術を必要とし、市場に投入するのに時間がかかる。

#### 【0007】

解読鍵をPLDの不揮発性メモリに記憶し、暗号化されたビットストリームをPLDにロードし、PLD内の鍵を用いてそのビットストリームを解読することも公知である。これにより、攻撃者は、PLDにロードされているときにビットストリームを读出することができず、PLDから電源が引かれたとき鍵を保持できる。そのような構成は、米国特許第5,388,157号においてAustinにより記載される。しかし、この構造は、ユーザの設計を攻撃のすべてのモードから保護しない。

#### 【0008】

設計保護に加えて、ユーザの中にはデータ保護を必要とする者もいる。彼らは、PLDが電源を失うとき失われるべきでないデータをPLD内に生成したかもしれない。そのようなデータを保護することが望まれる。

#### 【0009】

便利で確実かつ安全な設計保護方法が依然必要とされている。

#### 〔発明の概要〕

この発明は、無許可の使用およびデータ損失からPLDを保護するためのいくつかの構造および方法を提供する。

#### 【0010】

電源投入時にロードされなければならないスタティックRAMメモリによってPLDが構成される場合、構成データは、デバイスにロードされているときに保護されなければならない。先行技術におけるように、これは、構成データを暗号化しそれを集積回路装置の外部のメモリに記憶し、1つ以上の解読鍵をPLDにロ

ードし、電源が切られたときPLDにその鍵を維持し、鍵を用いて構成データを解読する解読回路をPLD内に含め、PLD内で解読された構成データを生成し、解読された構成データを用いてPLDを構成することにより達成される。

#### 【0011】

追加のセキュリティのため、不揮発性メモリを用いて鍵を保存するのではなく、この発明は好ましくは、PLDに接続される電池を用いて、電源がPLDから引抜かれたとき鍵を保存する。鍵を不揮発性メモリに記憶するPLDを取外し、PLDの覆いを取除き、不揮発性ビットのうちどれが論理1にプログラムされどれが論理0にプログラムされているのかを観察することが可能であるが、スタティックメモリセルにのみ記憶された鍵の内容を判断することは非常に困難であると考えられる、というのも鍵を記憶するためにさえも、鍵を記憶するメモリセルの電源が維持されなければならない、動作電源がPLDに対して連続している状態で、PLDの覆いを取除き層を除去しプローブを行わなければならないはずだからである。

#### 【0012】

##### PLDにロードされた設計を攻撃者が盗み得る態様

鍵が十分なセキュリティを提供しなければ、攻撃者は暗号化コードを解読し、鍵の値を判定してしまうであろう。周知のデータ暗号化規格DESは56ビットの暗号化鍵を用いており、高度なコンピュータで数時間のうちに解読されてしまい鍵を暴露してしまう。DESは、John Wiley & Sons, Inc. により発行された「応用暗号手法第二版：プロトコル、アルゴリズム、およびC言語でのソースコード」(“Applied Cryptography Second Edition: protocols, algorithms, and source code in C”) 著作権1996年の265~278頁においてBruce Schneierにより記載される。そのような周知の暗号化規格を用いることが望ましい場合、セキュリティを増強するために、構成データは、毎回異なった鍵を用いて何回も暗号化され得り、暗号化が繰返されるたびに約2<sup>56</sup>だけ暗号化コードを強化する。もしくは、それは、第1の鍵を用いて暗号化され、第2の鍵を用いて解読され、第3の鍵を用いて暗号化されても

よく、三重DES規格の一部である組合せであってもよい。他の暗号化アルゴリズムを用いてもよく、セキュリティは鍵にあるのでアルゴリズムを秘密にする必要はない。暗号化方法が対称的であるとき、暗号化に使用されるのと同じ鍵をPLDに記憶し逆の順で用いて解読を行なう。

#### 【0013】

複数の鍵を提供するPLDにおいて、使用されるべき鍵の数およびすべての鍵のアドレスが、暗号化されていないビットストリームで与えられるならば、攻撃者は、一度に1つ鍵を攻撃して鍵の値をより簡単に判断できてしまうであろう。そのような攻撃を回避するために、いくつかの鍵が使用されるか、および鍵がある組のうち最後の鍵であるのかまたはさらに以下に続くのかの表示を、ビットストリームではなく鍵の中に記憶することにより、さらなるセキュリティが達成される。

#### 【0014】

ビットストリームがPLDにロードされた後ビットストリームをリードバックするというオプションをPLDが提供する場合、攻撃者が用い得る別の方法は、このビットストリームをリードバックすることである。設計を攻撃するこの方法を回避するために、ある実施例では、リードバックを提供しかつ暗号化も提供するPLDは、暗号化が使用されてしまうとリードバック特徴を不能化する能力を含む。別の実施例では、リードバックする能力を提供するPLDは、リードバックされる前に構成データを暗号化する。

#### 【0015】

さらに、いくつかのPLDは、部分的構成（設計のいくつかの部分をロードするためにいくつかの構成アドレスが特定される）、および部分的再構成（既存の設計が消去されてから新しい設計データがロードされる）というオプションを提供する。PLDがこれらのオプションを提供する場合、攻撃者は、PLDを部分的に再構成し、設計の連続部分を見えるようにして、おそらくは設計全体を知ることが可能である。そのような攻撃を回避するために、ある実施例では、暗号化された設計がロードされたPLDの部分的構成および再構成は不能化される。別の実施例では、いくつかの構成アドレスが特定され得るが、アドレスは暗号化され



る。

#### 【0016】

攻撃のさらに別のモードは、PLDのセキュリティ状態を示すビットを反転しやうとすることである。動作電圧を低下または上昇させ、温度を変化させ、ノイズをある種のポートに与えることが想定される。そのようなビット反転から保護するために、PLDが暗号化されたビットストリームで作動しているとき、安全モードフラグがセットされ、ある実施例では、このフラグが設定解除されると、すべての構成データが消去される。デバイスが作動している間でも再構成を許可しない別の実施例では、構成データは、ビットストリームが送信される前に消去される。

#### 【0017】

攻撃の別のモードは、暗号化されたビットストリームの一部を再配置し、それらが非暗号化されるとき、設計者が意図しないPLDの見える部分に置かれるようにすることである。この再配置を防ぐために、暗号化および解読プロセスにおいてアドレス情報を用いて、暗号化されたビットストリームの一部を、設計者が意図したものとは異なるPLD場所へ送信することにより、それが意味を有さないデータに異なって解読される。暗号ブロック連鎖(CBC)は、この結果を達成する効果的な手段の1つである。暗号ブロック連作において、解読されたデータパケット(ブロック)は、排他的論理和関数を用いて次のデータブロックと組合され、その後次に次のブロックを解読し、そのため各データブロックごとの暗号化されたデータは、それに先行するあらゆるブロックおよびこれらのブロックの順番に依存する。データの同一のブロックは、それらに先行するデータブロックの値に依存して異なった値に暗号化される。このように、もしブロックの順序が変更されるならばビットストリームは正しく解読されない、というのも暗号化されたビットストリームが再配置される場所が後続のデータにスクランブルをかけるからである。さらに、初期CBC値を修正してデータのアドレスを組込み解読されたデータが特定の位置に置かれるようにして正しく解読を行なうことができる。

#### 【0018】

代替的に、PLDが、設計の一部が暗号化され一部が暗号化されないことを可能

にしているならば、攻撃者は、設計の暗号化された部分についての情報を读出そうとする暗号化されていない部分を暗号化された部分に加えるかもしれない。したがって、設計が完全に暗号化されるかまたは完全に暗号化されないことを許可するが、これらが混ざらないようにすることにより、さらなるセキュリティが達成される。さらにまた、ある実施例では、データが暗号化されているとき、構成データの単一の開始アドレスの後に単一のフルチップ構成のみを許可することにより、追加のセキュリティが提供される。

#### 【0019】

さらに、便利な検査およびデバックを可能にし、かつPLD製造者がその顧客（設計を製造しPLDを構成する設計者）と自由にやり取りすることを可能とするために、PLDは、動作の暗号化モードおよび非暗号化モードの両方を有し、暗号化モードで動作するとき、PLDへの構成データのロードを制御する構成ビットストリームの一部はまだ暗号化されない。

#### 【0020】

攻撃の別のモードとして、PLD製造業者が、構成データをロードするためのヘッダ情報およびアドレスを含む、構成ビットストリームフォーマットについての情報を自由に与え、使用された暗号化方法についての情報を与える場合、この周知の情報を暗号化することで、暗号化鍵が暴かれるかもしれない。そのような暴露は、実際の構成データのみを暗号化し制御情報を非暗号化のままにすることにより回避される。

#### 【0021】

PLD製造業者が、鍵メモリが安全モードおよび非安全モードの両方で使用されることを可能にしている場合、攻撃者は、鍵メモリを非安全モードにし鍵を读出すことにより、鍵を簡単に知ることができるかもしれない。そのような攻撃を回避するため、PLD製造業者は、鍵メモリが非安全モードに移されると、PLDにロードされたすべての鍵およびいずれの構成データも消去されるようにする回路を含める。

#### 【0022】

〔詳細な説明〕

図1は、FPGA10の先行技術の構造を示す。FPGAはプログラマブルロジック11を含み、プログラマブルロジック11は典型的には、(1)ルックアップテーブル組合せ論理関数生成器と、ルックアップテーブルの出力および他の値を記憶するためのフリップフロップと、プログラマブルロジックの論理能力を向上させるためのマルチプレクサおよび論理ゲートとを有する論理ブロックと、(2)信号をFPGAのまわりに経路付けるための経路およびプログラマブル相互接続点と、(3)経路とFPGAの外部ピンとの間で信号を駆動するための入力/出力ブロックとを含む。

#### 【0023】

FPGAはまた、経路選択トランジスタをオンにし、マルチプレクサを制御し、ルックアップテーブルを記憶し、入力/出力ブロックを制御するための構成メモリ12を含み、このすべては、FPGAを構成し設計者の所望する機能を行なうするためのものである。バス16は、構成メモリ12をプログラマブルロジック11に接続し、典型的には、FPGA全体にわたっている制御線の分布された組である。いくつかのザイリンクス・インコーポレイテッドの製品(たとえばXC6200)は、プログラマブルロジック11により構成ロジック14がプログラミング情報を構成メモリ12に送るバス17を含んでいる。そのような構造は、米国特許第5,705,938号においてKeanによって記載される。

#### 【0024】

FPGA10はさらに、JTAGポート20とインターフェイスし、特にFPGAが置かれる基板の検査を意図したJTAG論理ブロック13を含む。JTAG論理ブロック13は、IEEE規格1149.1のスーパーセットであるIEEE規格1532を実現する。JTAGは基板レベルでの設計のデバッグを可能にする。

#### 【0025】

最後に、FPGA10は、構成アクセスポート21で外部源15からの構成ビットストリームに応答し、JTAG論理ブロック13とインターフェイスする構成ロジック14を含む。構成アクセスポート21上のビットストリームは、ワード、ある実施例では32ビットのワードとして扱われる。通常ビットストリーム

の開始または開始付近でのワードのいくつかは、構成プロセスを設定するために用いられ、たとえば構成メモリフレームの長さ、および構成データの開始アドレスを含む。バス19は、構成ロジック14とJTAG論理ブロック13との通信を可能にし、JTAGポートを別の構成アクセスポートとして用いることができる。バス18は、構成論理ブロック14と構成メモリ12との間の通信を可能にする。特に、それは、メモリ12の構成フレームを選択するためのアドレスと、書込動作および読出動作を行なうための制御信号と、構成メモリ12にロードするかまたは構成メモリ12からリードバックするデータとを運ぶ。

#### 【0026】

構成論理ブロック14は、命令およびデータを受け、命令に従ってデータを処理する。これらの命令は、ビットストリームとして構成ロジック14に入る。命令、またはヘッダの後には通常、処理すべきデータが続く。図2aは、例示のビットストリーム構造を示す。ヘッダAは処理を特定し、単一のワード、すなわちデータAが後続することを特定する。ヘッダBは処理を特定し、この場合には掘すべき4ワードのデータが後続することを特定する。

#### 【0027】

図2bは、ザイリンクス・インコーポレイテッドから入手可能なバーテックス（Vertex）（R）において使用されるビットストリームにおける32ビットのヘッダワードのデフォルトのフォーマット（フォーマットタイプ001）である（バーテックス（R）は、この発明の譲受人であるザイリンクス・インコーポレイテッドの登録商標である）。このフォーマットは、フォーマットタイプ（001）を示す3ビットと、演算コードを特定するための2ビットと、構成論理レジスタアドレスのための16ビットと、ワードカウントのための11ビットとを含む。演算コードは、読出動作、書込動作、または演算なしを指定することができる。たとえば、00は演算なしを指定することができ、01は読出を指定することができ、10は書込を指定することができる。ワードカウントのための11ビットは、211ワードまたは2048ワードを指定することができる。図2cに示すように、ワードカウントがこれよりも大きければ、フォーマットタイプ001でのワードカウントビットは000000000000に設定され、フォ

フォーマットタイプ001のヘッダの後にはフォーマットタイプ2のヘッダが続く。フォーマットタイプ2は27ビットを用いてワードカウントを特定し、したがって $2^{27}$ ワードまたは268万ワードを特定することができる。

#### 【0028】

図2dは、パーテックス(R)ビットストリームのヘッダが構成ロジック14のレジスタにロードすることのできる制御情報の種類を示す。たとえば、構成論理レジスタアドレス0000を有するヘッダ(フォーマット001)は、次の32ビットのデータワードが巡回冗長検査(CRC)レジスタにロードされるべきことを明示する。(パーテックス(R)のデバイスは、16ビットの巡回冗長検査値を用いるので、いくつかのビットは0で埋められる。)ヘッダがアドレス0001を含む場合、構成メモリ12中のフレーム(列)を明示してデータを受けたり与えたりするために、次のデータがフレームアドレスレジスタにロードされる。

#### 【0029】

図2bに示す構成論理レジスタアドレス(16ビット)は、図2dの左側の列に示す4ビットの値を与え、この4ビットの値は、次の32ビットのデータワードを置くべき構成ロジック14(図1)中のレジスタの1つを選択する。フレーム長レジスタ(アドレス1011)は、構成データがロードされるフレームの長さを特定する。(フレーム長、または列高さは、PLDの寸法に依存する。より大規模なPLDは通常より高い列またはより長いフレームを有する。PLDに異なった構造を設けてデータワードをフレームに置くのではなく、ビットストリームのフレーム長を特定しフレーム長をレジスタに記憶することで、内部の構成ロジックは異なった寸法のPLDに対して同一のものとなる。)

リードバックに関しては、読出コマンドが演算コードフィールドに置かれ、フレームデータ出力レジスタがアドレスされ、その後にはワードカウントが続く(必要であればコマンドヘッダフォーマット2を用いる)。特定の数のワードが、フレームアドレスレジスタに特定されるアドレスから始まって構成メモリ12からリードバックされ、構成アクセスポート21またはJTAGポート20のいずれかでシフトアウトされる。(リードバックデータは、リードバック命令を発行した

ポートに返される)。

#### 【0030】

ビットストリームヘッダまたはヘッダの対にワードカウントを特定することで(図2bおよび図2c)、データワードがロードされるときカウントダウンするカウンタを設定する。多くの構成論理レジスタアドレスについて、ワードカウントは1である。ビットストリームヘッダが、構成データがロードまたはリードバックされていることを示す0010または0011の構成論理アドレスを有する場合、ワードカウントはもっと大きくなる。これは、図2cのヘッダフォーマット2が使用された場合である。フレームデータ入力レジスタ(アドレス0010)を介して構成メモリ12にロードされたデータ、またはフレームデータ出力レジスタ(アドレス0011)を介して読出されたデータは、設計データと呼ばれる、というのもこれは、FPGAに設計を実現させたり設計の状態を示したりするからである。他のレジスタデータは制御データである、というのもそれは、ロジックが構成またはリードバックされている間に構成ロジックがどのように振る舞うかを制御するからである。

#### 【0031】

バーテックス(R)デバイスの構成についての一層の詳細は、95124 カリフォルニア州、サン・ノゼ、ロジック・ドライブ、2100のザイリンクス・インコーポレイテッド(この発明の譲受人)により2000年10月9日に出版された「バーテックス構成ガイド」(“Virtex Configuration Guide”)に見出すことができる。

#### 【0032】

構成ロジック14は典型的には、入来する構成ビットストリームに対して巡回冗長検査を行ない(Ericksonの米国特許第5,321,704号または上に参照された「バーテックス構成ガイド」の39から40ページを参照された)、構成されている部分のフレーム長および構成データのワードカウントを示すヘッダビットを讀出し、構成データをどこにロードすべきかを識別するアドレス情報を讀出し、構成データのフレームを収集してそれらをアドレスに示される構成メモリ12の列にロードする。構成ロジック14はまた、構成メモリ12か

ら外部の場所への構成データおよびフリップフロップ値のリードバックを制御する。ザイリンクス・インコーポレイテッドから入手可能なパーテックス（R）FPGAにおいては、リードバックは、JTAGポート20または構成アクセスポート21のいずれかを介して行なうことができる。

#### 【0033】

構成ロジック14はまた、プログラマブルロジック11から構成データを受けることができる。FPGAの一部がFPGAの別の部分を構成する先行技術のFPGA構造についてのさらなる情報は、Keanの米国特許第5,705,938号に見られる。パーテックス（R）アーキテクチャと同様のFPGAのアーキテクチャについての一層多くの情報は、Youngらの米国特許第5,914,616号に見出せる。この発明の譲受人であるザイリンクス・インコーポレイテッドから入手可能なパーテックス（R）製品で用いられるビットストリームのフォーマットは、95124 カリフォルニア州、サン・ノゼ、ロジック・ドライブ、2100のザイリンクス・インコーポレイテッドから入手可能で2000年10月4日に発行された「パーテックスFPGAシリーズの構成およびリードバック」（“Virtex FPGA Series Configuration and Readback”）と題するアプリケーションノートXAPP138に記載される。

#### 【0034】

##### 解読付きのPLD

図3は、この発明のある実施例に従うFPGA（PLDの1タイプ）のブロック図を示す。いくつかの要素は図1に示すのと同じであり、同じ参照番号が付与されており、再び説明しない。さらに、図3は、拡張構成論理ユニット29と、解読器24と、鍵メモリ23とを含む。図3は、鍵メモリ23がバス25でJTAGアクセスポート20からロードされる実施例を示す。他の実施例では、鍵メモリ23は、別のポートを介してロードされる。バス25は、書込および読出動作を行なうためのデータ、アドレスおよび制御信号を選び、JTAGポート20からの解読鍵のプログラミングを可能にする。ある実施例では、バス26は、構成ポートからの鍵のプログラミングを可能にする。別の実施例では、バス26が削

除される。さらに別の実施例では、バス26が存在しバス25が削除される。ここにさらに記載する実施例では、バス26は、鍵メモリ23から構成ロジック29へセキュリティデータを運ぶ。ある実施例では、バス27は、構成ロジック29から解読器24へ暗号化された構成データを選び、解読された構成データを構成ロジック29に戻す。バス28は、解読器24が鍵にアクセスしてデータを解読することを可能にする。図3の構造に暗号化されたデータがロードされているとき、ビットストリームがロードされているのをモニタする攻撃者は、暗号化されたビットストリームのみを受信し、この方法によりユーザの設計を知ることができない。

#### 【0035】

##### 部分的に暗号化されたビットストリーム

この発明の別の局面に従えば、ビットストリームは、2つの部分、すなわち暗号化できるまたはできないユーザの設計を表わすデータ部と、ビットストリームのロードを制御する（たとえば、ビットストリームの連続部分がロードされるPLDの列のアドレスを与え、ローディング動作の信頼性を検査するための巡回冗長検査（CRC）コードを与え、暗号ブロック連鎖（CBC）、すなわち暗号化されたデータの発生頻度から解読されたデータを導出することができる「辞書攻撃」を防ぐ技術、の開始番号を与える）制御部とを含む。この発明の好ましい実施例では、データ部は暗号化されてもよいが制御部は暗号化されない。これは追加のセキュリティを与える、なぜならPLD製造業者はビットストリームの制御特徴を自由に記述する必要があり、この比較的良好に知られた制御情報が暗号化された場合、攻撃者は、この情報を解読しこの情報を用いてビットストリーム全体を解読できるかもしれないからである。さらに、ビットストリームの制御部を暗号化されない状態にしておくことにより、PLDは情報を使用しやすくなる。

#### 【0036】

別の実施例では、構成データがロードされるアドレスの順序が、設計を解析するに当り攻撃者に役立つかもしれない場合に用いられ、構成データのアドレスも暗号化されるが、構成ビットストリームの他の制御情報は暗号化されないままである。



## 【0037】

ビットストリームフォーマット

図4a～図4dは、図2a～図2dに示す先行技術の製品の構成ロジック14のフォーマットおよびレジスタと比較した構成ロジック29のビットストリームフォーマットおよびレジスタの違いを示す。図4aに示すように、ビットストリームはさらに、ヘッダワードと、後続のデータワードとを含む。典型的な構成では、いくつかの構成データワードはレジスタにロードされ、その後に、暗号化された構成データが始まる。図4aは、3つのヘッダワード、すなわちヘッダA、ヘッダBおよびヘッダCの各々の後に、3つの暗号化されていない制御データワード、すなわちデータA、データBおよびデータCが続く例を示す。（実際の構成では、3つより多い制御データワードが設けられる可能性が高い。）次に、ヘッダDが、暗号化された構成データが後に続くことを明示し、その後には、暗号化された構成データの多数のワード、すなわちデータ1D、データ2D、データ3Dなどが続く。このデータが暗号化されていることを強調するために、これらのワードは図4aにおいて斜線が付けられている。

## 【0038】

図4bおよび図4cに示すように、4つの演算コードが追加される。演算なしの値00、解読なしの読出および書込の01および10に加えて、新しい値11は、書込が解読つきであることを明示する（解読が用いられることや、それが演算コードにより特定されることを明示するために、どのようなコードまたはどのような方法が用いられるかは重要でない）。オプションの暗号化および解読が可能とされ示されることにより、設計者がこのオプションを利用できることだけが重要である。図4dの実施例では、2つの新しい構成論理レジスタが追加される。アドレス1100および1101で示されるのは、暗号ブロック連鎖（CBC）開始値を保持するレジスタと、初期暗号化鍵のアドレスとである。

## 【0039】

オプションの暗号化

この発明の別の実施例に従えば、PLDはビットストリームの暗号化されたデータ部と暗号化されていないデータ部との両方を受付けることができる。ビットス

トリームの制御部は、ビットストリームデータのデータ部が暗号化されているかどうかを示す。ビットストリームのデータ部が暗号化されている場合、それはPLD内で解読器へと進路を変え、解読後にPLDを構成するために用いられる。暗号化されていない場合、それは進路を変えず、PLDを構成するために直接使用される。

#### 【0040】

ビットストリームを暗号化しないことが好ましい場合がいくつかある。設計をデバックする際に使用されるある種のテスト動作は、構成情報をリードバックする必要がある。暗号化ステップが行なわれなかったならば（特に設計者が暗号化が問題と関係あるかどうかを判定しようとする場合）構成上の問題を診断することはより簡単である。また、数人の設計者が、PLDの幾つかの部分で実現されるコードを書いており、PLDの異なった部分が別々のときに構成される場合、ビットストリームのすべての部分が見える状態にし、PLDが部分的に再構成されることを可能にする必要があるかもしれない。

#### 【0041】

図5aおよび図5bは、最初に暗号化されておらず次に暗号化される同じ設計を表わす例示のビットストリーム部を示し、この発明のある実施例における暗号化されていないビットストリームと暗号化されたビットストリームとの違いを示す。実際のビットストリームは、図の右側の0および1を含み、左側の文を含まない。左側の文は、右側のビットの意味を説明するために設けられている。これらのビットストリーム部は、図4b～図4dに示すコマンドを用いる。図5aの暗号化されていないバージョンと図5bの暗号化されたバージョンとの違いを強調するために、この違いは太字で示してある。

#### 【0042】

図5aを見ると、ダミーワード（すべて1と解釈される一定のハイの信号である）および1および0の特定のパターンを有する同期ワード（sync word）の後の、次のワードは、10の演算コードを有するタイプ001であり、00000000000010000のアドレスおよび0000000000001のワードカウントを有する。したがって、このワードは、コマンドレジスタCMDを

アドレスし、1ワードがそこに書込まれることを特定する。図5aにはビットストリームの左側に、このワードがType 1であることを示しかつwrite 1 word to CMDを示す注釈が付けられている。次のワード111は、コマンドレジスタCMDに与えられるべきデータであり、CRC（巡回冗長検査）レジスタをリセットする（好ましい実施例では、PLDは、ビットストリームがロードされているときにビットストリームからCRC値を計算するための、米国特許第5,598,424号においてEricksonによって記載されるような図示しない回路を含み、正しくないビットがロードされてしまいかねないビットストリーム電圧のグリッチから保護する）。次に、ヘッダワードは、フォーマットがやはりタイプ1であることを明示し、1ワードをフレーム長レジスタFLRに書込むことを明示する。その後続くデータワード11001は、フレーム長（25ワード）を特定する。同様に、フレームアドレスレジスタFARに書込まれるべきワードを特定するヘッダを含む、いくつかの追加のヘッダおよびデータワードが続く。この場合には、後に続くデータワードは、データがアドレス0で始まることを示す。最後にこれらのレジスタがロードされてしまった後、コマンドはデータをフレームデータ入力レジスタFDRIに書込むようになり、多くのデータが書込まれるので、ワードカウントは000000000000として与えられ、タイプ2のヘッダは、10530ワードがFDRIレジスタに書込まれることを特定する。これは、PLDが構成されるようにする実際の設計データである。したがって、ビットストリームにおける次の10530ワードは設計データである。最後に、データが正しくロードされたことを保証するために、構成データを発生させたデバイスによって計算されたCRC値がロードされ、PLDによって計算されたCRC値と比較される。構成が完全なものであることを示し、かつPLDを演算モードに移すために、さらなるコマンドおよびデータがロードされる。

#### 【0043】

図5bは図5aと同様であるが、データおよび注釈が太字で示されている点のみが異なる。図5bにおいて、データは暗号化されており、追加のコマンドを用いて初期鍵アドレスを与え2ワード（64ビット）をCBC（暗号ブロック連鎖）

レジスタに書込む。次に、タイプ1ヘッダは、演算コード11を含み、フレームデータ入力レジスタFDR1に書込まれる前にデータが解読されることを示す。その後にはタイプ2ヘッダが続き、やはり演算コード11を有しており、10530ワードが解読されデータ入力レジスタFDR1に書込まれる旨の命令を与える。10530個の暗号化されたデータワードが次に続く。次に、CRCワードが続き、(暗号化された)データが正しくロードされたことを確認する。最後に、追加のコマンドおよびデータが送信され、すべて正しいならばPLDを演算モードにする。

#### 【0044】

##### 解読プロセス

図6は、ある実施例においてオプションの解読が達成される態様を示す。図6は、構成ロジック29と、解読器24へ至るバス27および28との詳細を示す。バス27は以下を含む：

- ・構成ロジック29のレジスタアドレス1101 (図4d) から取得された3ビットの初期解読鍵アドレス「Init\_key\_addr」
- ・64ビットの修正された暗号ブロック連鎖値「modCBC」。この値は、構成ロジック29のレジスタアドレス1100 (図4d) から取得された64ビットのCBC値の下位ビットをレジスタ0001に特定される22ビットのフレームアドレス値と置換することにより形成される。

#### 【0045】

- ・ビットストリームから取られ、解読されたデータをロードするための64本の回線「Encrypted\_data」
- ・解読器24により生成された解読データを構成ロジック29に返すための64本の回線「Decrypted\_data」
- ・データが「Encrypted\_data\_lines」上にあることを解読器24に示し、かつ解読器24が解読を開始できることを示す信号のための回線「Enc\_data\_rdy」
- ・64ビットのワード上の解読が完了し「Decrypted\_data」上で利用可能であることを構成ロジック29に示す信号のための回線「Dec\_data

t a\_\_r d y J

・ 解読器 24 により使用され、構成ロジック 29 に構成を中止させ、それにともない、たとえば鍵がある組の最初、中間または最後のものであるかを表わす鍵メモリ中のビットにより特定されるとおりに鍵が使用されていないとき、状態レジスタをセットする B a d\_\_k e y\_\_s e t。図 4 d に示す実施例では、状態レジスタはアドレス 0111 にあり、B a d\_\_k e y\_\_s e t エラーは、論理 1 をビットの 1 つに記憶することにより示される。

【0046】

バス 28 は、以下からなる：

- ・ 最初にはバス 27 から与えられる鍵アドレスであるが、新しい鍵が使用されるたびに更新される、鍵アドレスのための 3 本の回線
- ・ 解読鍵のための 56 本の回線、および
- ・ 解読鍵が最初、中間、最後または使用される唯一の鍵であることを示すための 2 本の回線。

【0047】

#### 設計再配置の防止

暗号化されたビットストリームにおける設計に対する攻撃の可能性の 1 つに、暗号化されたビットストリーム中のフレームアドレスレジスタ（開始アドレス）を変更して、それが解読されたとき、F P G A が使用されているときに見える F P G A の一部にロードされるようにすることがある。いくつかの設計では、ブロック R A M の内容は見える。すべての設計において、入力／出力ポートの構成は見える状態になっており、そのため、構成ビットを判断することができる。したがって、設計の連続部分が F P G A の見える部分に移動された場合、F P G A が適切に機能しなかったとしても、攻撃者は、再配置を繰り返すことで、暗号化されていないビットストリームの内容を知ることができる。

【0048】

設計の再配置を防ぐために、ある実施例では、D E S 規格とともに用いられる暗号ブロック連鎖方法によって使用される初期値が修正される。図 7 a および図 7 b は、この発明に従って修正された、三重 D E S アルゴリズムの暗号化部と解読

部とをそれぞれ示す。標準の暗号ブロック連鎖方法は、開始番号（設計者が供給したものであっても、ランダムに生成されたものであってもよい）と暗号化されるべきデータの最初のワードとの排他的論理和を取ることで暗号化プロセスを開始する。この発明に従えば、乱数の一部が、アドレス情報、この例では構成メモリ12においてデータがロードされる最初のフレームの22ビットのアドレスと置換される。開始CBC値、すなわち64ビットの数は、 $x$ と標識付けされた最下位ビットが $y$ と標識付けされたフレームアドレスと置換され、データがロードされるアドレスに依存する修正された64ビット値を生成する。この修正されたCBC値は構成情報ワード1の最初のワードと排他的論理和が取られる。暗号化アルゴリズムを用いて最初の暗号化されたワード、すなわち暗号化ワード1を生成し、これはビットストリームに与えられる。図7aは、外部暗号ブロック連鎖を有する三重暗号化アルゴリズムを示し、これは、第1の鍵を用いる暗号化ステップ $enc_1$ と、その後続く第2の鍵を用いる解読ステップ $dec_2$ と、その後続く第3の鍵を用いる暗号化ステップ $enc_3$ を含む。この第1の暗号化されたワード、すなわち暗号化ワード1は、第2の暗号化されていないワード、すなわちワード2との排他的論理和が取られ、暗号化プロセスを繰返して暗号化されたワード2を生成する。すべての構成データが暗号化されてしまうまで排他的論理和連鎖は続く。

#### 【0049】

図7bに示すように、PLDは、解読されたワードを得るために逆のプロセスを行なわなければならない。上述の暗号化シーケンスについて、解読シーケンスは、鍵3を用いる解読ステップ $dec_1$ と、次に鍵2を用いる暗号化ステップ $enc_2$ と、次に鍵1を用いる解読ステップ $dec_3$ とである。重要なことは、解読ワード1を生成するための初期値の一部は、暗号化についても解読についても同じフレームアドレスを用いることである。ビットストリームではなく、PLDは、フレームアドレスレジスタに記憶されたフレームアドレスから修正されたCBC値を生成するが、これはまた、構成データがロードされる構成メモリ12のフレームを特定するためにも使用される。したがって、攻撃者が、データがロードされるフレームアドレスを変更すると、修正されたCBC値がそれに伴って変

化し、構成データは正しく解釈されない。

#### 【0050】

排他的論理和ステップは、暗号化される前は設計者のビットストリームにあった元々のデータを生成する。たとえば、元のワード1＝解読ワード1である。この解読された構成データは、バス27（図3）で構成ロジック29に送られる。

#### 【0051】

#### 構成ロジック29

構成ロジック29は、オプションの暗号化をサポートする構造と、設計再配置および単一鍵攻撃を防止するための構造とを含む。図6に示すように、構成ロジック29は、保持レジスタ292と、制御ロジック291と、構成レジスタ（FDR1、FAR、CRCおよびinitCBCが示される）と、解読器24インターフェイスマルチプレクサ294および295と、64ビットのアッセンブリレジスタ297と、レジスタ298および299（構成アクセスポート221とインターフェイスする）とを含む。64ビットのシフトレジスタ299は、1ビット幅のデータのための単一のピンであっても、8ビット幅のデータのための8個のピンであってもよい構成アクセスポート21からデータを受ける。レジスタ299がいっぱいになるまでこのデータは64ビットのシフトレジスタ299にロードされる。そして、これらの64ビットは、好ましくは、64ビット転送レジスタ298に並列にシフトされる。そこから、マルチプレクサ296bが右側および左側の32ビットワードを交互に選択し、マルチプレクサ296aが、制御線Mによって制御されるように、保持レジスタ292へまたはアッセンブリレジスタ297の高位部分と低位部分とへ交互に、データを一度に32ビットずつ移動する。ビットストリームのロードが始まると、回線Mおよび図示しないクロック信号により、マルチプレクサ296aおよび296bは、64ビット転送レジスタ298から保持レジスタ292にデータを移動する。そこから、これらのワードは制御ロジック291に与えられる。ワードがヘッダである場合、制御ロジック291はこのワードを解釈する。演算コードが、後続するデータが暗号化されずに書込まれるべきことを示す場合、制御ロジック291は、レジスタを選択するためにバスGにアドレスを与え、回線Lに信号を与えてマルチプレクサ29

4にバスBからバスDに接続させ、これに続くワードをバスBに与える。次のクロック信号(クロック信号は図示しない)で、バスD上のデータはアドレスされたレジスタにロードされる。図4dに示すすべてのレジスタは、このようにロードされ得る。初期暗号ブロック連鎖値をロードするためのinitCBCレジスタは、64ビットのレジスタであり、図5bに示し上述したように2つの連続した32ビットワードを受ける。

#### 【0052】

(1) initCBCレジスタに記憶された元のCBC値と、(2) FARレジスタに記憶された初期フレームアドレスとから形成された、修正されたCBC値は、解読器24に利用可能となる。ある実施例では、FARレジスタ中の初期フレームアドレスは、32ビット以下を使用するが、initCBC値は64ビットを用いる。図6の実施例では、修正されたCBC値を供給する64ビットのバスは、フレームアドレスレジスタFARからの22ビットと、initCBCレジスタからの42ビットとを含む。この発明によって提供されるセキュリティに重要なことには、この値は構成データがロードされるところに依存する。攻撃者が、FARレジスタの内容を変えることにより、暗号化されたデータを異なった場所にロードしようとするならば、解読器24に供給されるmodCBC値も変化してしまうだろう。

#### 【0053】

構成データの多数のワードを解読するための演算コードコマンドが制御ロジック291によって受けられると、解読プロセスが始まる。制御線Mにより、マルチプレクサ296aは、転送レジスタ298からアッセンブリレジスタ297に至るバスAにデータを与える。制御バスHは、暗号化データレジスタ297の高位[31:0]部および低位[31:0]部とにバスAを交互に接続して、解読されるべき64ビットワードを形成する。制御ロジック291は次に、Encmdata\_rdy信号をアサートし、これにより解読器24はレジスタ297中のデータを解読する。

#### 【0054】

解読を行なうために、解読器24は、鍵アドレスKey Addrをバス28で



鍵メモリ23(図3)に与える。これにより、鍵メモリ23はそのアドレスの56ビットの鍵を56ビット鍵線に返す。また、鍵メモリ23はそのアドレスの鍵データに記憶された2つの追加のビット「Order」を返す。第1の解読鍵について、これらの2つのビットは、これが第1の鍵または唯一の鍵であることを示さなければならない。そうでない場合、解読器24はBad\_key\_set信号をアサートし、これにより制御ロジック29は構成動作を中止する。これらの2つのビットが、鍵が第1の鍵または唯一の鍵であることを示す場合、解読器24は、たとえば(Schneierにより同書に記載される)周知のDESアルゴリズムを用いて解読を行なう。鍵が唯一の鍵でなければ、解読器24は鍵メモリ23中の次のアドレスの鍵を獲得し、それが中間または最後の鍵かを2つのOrderビットが示すかどうかを確認する。そうでなければ、Bad\_key\_set信号がアサートされ構成が中止される。もしそうであれば、解読が行なわれる。それが中間の鍵であるならば、解読の別の一巡が行なわれる。それが最後の鍵であるならば、解読器24は、解読されたワードと値modCBCとの排他的論理和関数を形成する。解読器24は次に、得られた値を64ビットのDecrypted\_dataバスに与え、Dec\_data\_rdy信号をアサートする。これにより、制御ロジック291は信号を制御線Kに与えて、マルチプレクサ295が64ビットワードを2つの連続した32ビットワードに分割するようにする。制御ロジック291は信号を回線Lに与えて、マルチプレクサ294が解読されたデータの32ビットワードをバスDに転送するようにする。制御ロジック291はまた、アドレス信号をバスGに与えてフレームデータ入力レジスタFDR1をアドレスする。次のクロック信号が、解読されたデータをバスEに移動させ、それはフレームレジスタにロードされ、フレームレジスタが一杯になると、FARレジスタに示されるアドレスで構成メモリ12に最終的にシフトされる。

#### 【0055】

modCBC値は、解読動作において1回のみ使用される。暗号化されたデータの後続の64ビットワードが解読され、先に解読されたデータを用いて連鎖される排他的論理和演算を行なう。(FARレジスタに記憶される値も、フレームアド

レスを選択するために1回のみ使用される。したがって、フレームが一杯になるたびにフレームアドレスはインクリメントされるだけである。)

#### 動作の流れ

図8は、構成ロジック29および解読器24によって行なわれる動作の流れを示す。構成ロジック29は、ビットストリームヘッダをロードし、対応するデータを図4bに示す構成論理レジスタに置くことにより、ビットストリーム長を判断することを含むステップ70で始まる。ステップ71で、起動シーケンスの後続の部分として、構成ロジック29は、最初の構成メモリアドレスを読み出す。ビットストリームフォーマットは、暗号化が用いられているかどうかを示す演算コードを含むことを想起されたい。ステップ72は、演算コード値により分岐する。暗号化が用いられていない場合、この処理は図8の左側部分に示すとおりである。暗号化が用いられている場合、処理は図8の右側に示されるものである。暗号化がない場合、ステップ73で、構成ロジック29は、カウンタをビットストリームワードカウント(図4c参照)に等しくセットする。ステップ74で、構成データの32ビット(1ワード)が構成メモリ12のアドレスされたフレームに送られる。カウンタが終了したことをステップ75が示す場合、ステップ76でカウンタがデクリメントされ、構成データの次の1ワードが構成メモリ12に送られる。カウンタが処理を終了した場合、構成ロジック29は、最終巡回冗長値を読み出してビットストリームの最後の値と比較しビットストリームのロードにエラーがあったかどうかを判定することを含むクリーンアップ動作を行なう。

#### **【0056】**

ビットストリームが暗号化されていることをステップ72が示す場合、カウンタにはワードカウントがロードされ、ステップ81で、この処理は、初期鍵アドレスを鍵アドレスレジスタ293(図6)から解読器24にロードする。

#### **【0057】**

ステップ82で、暗号化された構成データの2ワード(64ビット)が解読器24にロードされる。ステップ83で、アドレスされた鍵が解読器24にロードされる。ある実施例では、64ビットの数が解読器24にロードされる。この64ビットの数は、56ビットと、それが最初の鍵か、中間の鍵か最後の鍵かまたは

唯一の鍵であるかを示す2ビットと、未使用であってもパリティのために使用されても別の目的のために使用されてもよい、いくつかの他のビットとを含む。別の実施例では、64ビットの鍵データは、それが最後の鍵であるか否かを示す単一ビットを含む。さらに別の実施例では、64ビット鍵データが次の鍵のアドレスを含むので、鍵はシーケンシャルな順序で使用されなくてもよい。別の実施例では、追加ビットが存在せず、鍵データは64ビット未満を用いる。さらに別の実施例では、鍵ではなくビットストリームが、いくつかの鍵が使用されるべきかを示すが、これは、安全性が低いと考えられる。というのも攻撃者が、いくつかの鍵が用いられるかがわかり単一鍵攻撃を行なって、一度に1つの鍵を解読することができるからである。これに対し、いくつかの鍵が使用されるべきかを示す鍵を用いれば、この情報は攻撃者に漏れない。

#### 【0058】

ステップ84で、解読器24は、たとえばDESアルゴリズムを用いて56ビットの鍵で64ビットのデータを解読する。DESアルゴリズムは、Bruce Schneierによる上述の本の265から278頁に記載されている。たとえば、高度暗号化規格AESなどの他の暗号化アルゴリズムを用いてもよい。他のアルゴリズムはより多くの鍵ビットを必要とするかもしれない。たとえば、AESは128から256ビットの鍵を必要とする。

#### 【0059】

ステップ85は、より多くの鍵が使用されるかどうかを判断する。鍵が最初、中間、最後または唯一の鍵であるかを示す2ビットを調べて、これが最後の鍵であるかどうかを判定し、そうでなければ、鍵アドレスがインクリメントされ、解読器24はメモリ23中の次の鍵をアドレスする。

#### 【0060】

最後の鍵が使用された後、ステップ87で、レジスタFARおよびinitCB Cを組み合わせることから64ビット値として図6に示す修正されたCBC値と、ステップ87で得られた解読された値との排他的論理和がとられる。ある実施例では、CBCレジスタにロードされる64ビットの乱数のうち22ビットが、ビットストリームの始まりのフレームアドレスと置換される。この暗号化処理の目

標は、64ビットの暗号化された値のあらゆる桁を、鍵に加えてすべての前のビットの関数とすることである。CBC値を最初のアドレスと組合せることの目標は、ビットストリームが意図された開始アドレスと異なるアドレスにロードされる場合、解読された値が変わるようにすることである。ステップ87は、この目標の両方を達成する。次に、新しいCBC値が記憶される。図6に示すFARレジスタおよびCBCレジスタに記憶されても、または解読器24にある別のレジスタに記憶されてもよい。

#### 【0061】

ステップ88で、この解読された構成データはバス27（図3）で構成ロジック29に送られる。構成ロジック29は、更新された巡回冗長検査値を計算し、ロード処理の終りにCRCレジスタに記憶された巡回冗長値と比較する。構成ロジック29が暗号化を用いるように設定されているならば、構成ロジック29のマルチプレクサは、この解読された構成データを構成メモリ12のアドレスされた列に転送する。

#### 【0062】

ステップ89で、カウンタが検査され、もし終了していなければ、ステップ96でカウンタはデクリメントされ、処理はステップ82に戻り、ここで次の64ビット（2ワード）がビットストリームからロードされる。

#### 【0063】

最後に、カウンタが終了したことをステップ89が示す場合、ステップ90で、ビットストリーム中のCRC（巡回冗長検査）値が、ビットストリームがロードされるときに計算されたCRC値と比較される。これらの値が一致するならば、構成は完成しておりFPGAは動作に入る。これらの値が一致していなければ、ロードの際のエラーが発生しており、構成処理全体が中止される。

#### 【0064】

#### 鍵の順序の評価—単一鍵攻撃の防止

図9は、鍵の順序を評価するために解読器24によって実現される状態マシンを示す。状態マシンは、Enc\_data\_ready信号が活性化されるまで状態S1にある。この信号は、解読が開始可能で決定状態Q1に移ることを示

す。決定状態Q1では、解読器24は、バス27上のアドレスInit\_Key\_addrにより特定されるアドレスをバス28に与え、鍵および鍵の順序をリードバックし、2ビットの鍵順序データから、鍵が最初または唯一の鍵であるかを判定する。そうでなければ、解読器24は、Bad\_key\_set信号を制御ロジック291に送り、制御ロジック29に構成を中止させる。アドレスが最初または唯一のものであれば、解読器24は状態S3に入り、データを解読する。そして、状態マシンは決定状態Q2に入り、鍵が最後または唯一のものかを判断する。もしそうであれば、解読が完了し状態S4で解読器24は解読されたデータを構成ロジック29に返す。そうでなければ、状態S5で、解読器24は鍵アドレスをインクリメントし、新しい鍵を獲得する。状態マシンは、次の鍵が中間または最後の鍵であるかを判断するための質問Q3を尋ねる。そうでなければ、状態S2は構成を中止させる。鍵が中間または最後ののであれば、状態マシンは状態S3に戻りデータを再び解読する。別の実施例では、状態S4において、解読器24は、CBC値と解読されたデータとの排他的論理和をとるステップも行なう。

#### 【0065】

鍵の中に鍵の順序を記憶することの利益は、攻撃者が単一鍵攻撃を実現できないことである、というのも、攻撃者は、解読を行なう際、解読器24が（設計者によって意図される）鍵メモリ23によって特定される鍵のすべてを用いることを防ぐことができないからである。鍵の順序がPLD内の鍵データ内に記憶されるので、第2の質問Q2および第3の質問Q3を尋ねて単一鍵攻撃を使用する攻撃者から保護する必要はない。しかしながら、有利には、鍵をロードする設計者または基板検査師は、3つの質問のすべてを尋ねて各鍵がロードされる際に確実に正しく標識付けされるようにする。

#### 【0066】

ある実施例では、解読器24は、解読—暗号化—解読シーケンスを備える三重DES規格を用い、別の鍵が使用されるたびにアルゴリズムを（ほんの少しだけ）交互にする。そのような組合せは、ANSI X9.52 1998三重DES規格に従っている。別の実施例では、毎回解読が使用される。

## 【0067】

鍵メモリ23

図10aに示す回路は、3つの構成要素、すなわち電池供給スイッチ22と、制御ロジック23aと、鍵レジスタ23bとを含む。制御論理回路23aおよび鍵レジスタ23bは図3の鍵メモリ23を含む。図10aの実施例では、鍵レジスタ23bは6個の64ビットワードを含む。もちろん、他の鍵メモリの寸法がこれに代えて使用されてもよい。他の実施例では、鍵メモリ23に記憶される鍵は6つよりはるかに多くてもよく、使用される鍵のアドレスを与えるためには3ビットより多くが必要とされる。鍵レジスタ23bの電源は、ラインVSWITCHにより電池供給スイッチ22から来る。鍵メモリ供給電圧VCCIが不十分が存在していない場合、電池供給スイッチ22が電池バックアップ電圧VBATTをVSWITCHラインに与え、これによりVSWITCHは正の電圧を維持する。

## 【0068】

この実施例では、各鍵レジスタは64個のメモリセルを有する。各セルはライトイネーブル信号WEを受け、これは高レベルのとき、データがセルに書込まれ、低レベルのときセルのデータが保持される。あるレジスタにおけるセルは共通のライトイネーブル信号WEを有する。PLD電源電圧(VCCIと異なる)がなくそのためWE信号が活性に駆動されない場合、T1などの弱いプルダウントランジスタがWE信号をプルダウンして、いずれの鍵メモリレジスタもアドレス不可能とし、いずれのメモリセルも妨害されないようにする。

## 【0069】

ある実施例では、PLDのJTAGポートは、解読鍵をPLDにロードするために用いられる。メモリセル電源電圧は、通常動作時にはVCCIのデバイス電圧レベルにあり、ある実施例では、このレベルは3.0と3.6ボルトとの間である。JTAGポートに与えられる信号は幾つかの異なった電圧であってもよい。また、いくつかの異なった内部電圧があってもよい。したがって電圧変換が必要である。この電圧変換はメモリセルにおいて行なわれる。メモリセルの詳細は図10bに示される。インバータI1およびI2を含むラッチには、VSWITCH

Hによって電源が供給され、このためデバイス電源電圧VCCIがあってもなくても電源が供給される。WE信号および反転されたデータ信号data\_bはどちらも1.5ボルトのレベルで作用する。これらの信号は、NMOSトランジスタT4、T5およびT6を駆動し、インバータI3を介して(1.5ボルトの電源電圧も用いる)トランジスタT7を駆動する。図10bは、WEが低レベルのときトランジスタT4およびT5がオフであることを示し、インバータI1およびI2を含むラッチの内容が保持される。WEが高レベルのとき、インバータI1およびI2の一方が低レベルになり、新しいデータをラッチにロードする。

### 【0070】

制御論理回路23aはJTAGバス25(図3にも示す)から信号を受ける。JTAGバス25は、書込、読出、安全モードを設定するための制御信号ならびにデータバスおよびアドレスバスを含む。このインターフェイスは、IEEE1532JTAG規格に準拠している。鍵メモリ23がJTAGバス25を介してアクセス可能となる前に、セキュリティ状態(バス26)が非安全モードにされるが、これは、ISC\_\_PROGRAM\_\_SECURITY命令(図10aを参照)を用いて鍵データバスのビット0に論理1を与えることで行なうことができる。鍵メモリ23は、IEEE1532規格のISC\_\_PROGRAM命令およびISC\_\_READ命令を用いてJTAGバス25に書込まれJTAGバス25から(検証のために)読出される。制御ロジック23aは、JTAGバス25から3ビットのアドレス信号ADDRをデコードするデコーダを含み、ISC\_\_PROGRAM命令がJTAGバス25に現われる場合、書込ストロブ線ws\_\_b[5:0]のうちアドレスされたものに低レベルになるパルスを生じ、またはISC\_\_READ命令がJTAGバス25に現われる場合、読出選択線rsel[5:0]のうちアドレスされたものに高レベルの信号を生じ、6個の64ビットワードのうちの1つは、高レベルの信号を6本の読出選択線rsel[5:0]の1本に与え、読出マルチプレクサ23dが選択されたワードを64本の出力線q[63:0]に与えることにより、読出すことができる。書込選択線または読出選択線のうち1つのみが一度に選択される。どの読出選択信号もアサートされない場合、高レベルのpark\_low信号により、64個のトラン

レジスタ23eが64本の線q[63:0]をブルダウし、これらの回線が浮遊するのを防ぐ。

#### 【0071】

鍵メモリ23が非安全モードで動作している場合、64ビットワードを鍵レジスタ23bからJTAGバス25に読出すことができ、FPGAの外部で値を調べることができる。レジスタ23b中の選択された64ビットワードのうち56ビットをDES解読のための56ビットの鍵として用いることにより、FPGAをこの非安全モードでテストすることができる。ある実施例では、鍵メモリ23が非安全モードにあるとき、設計がロードする前に暗号化されていたとしてもユーザの設計のリードバックは可能である。これにより、設計者は暗号化された設計でもテストおよびデバッグすることができる。鍵セキュリティ状態の通信はバス26を介している(図3も参照)。

#### 【0072】

値が鍵レジスタ23bに書込まれバス25からの読出動作で確認された後、制御ロジック23aは、ISC\_PROGRAM\_SECURITY命令を用い、IEEE1532規格の一部である64ビットの鍵データバスのビット0に論理0を与えることにより安全モードに置かれる。安全モードでは、いかなる鍵へのアクセスも許されない。

#### 【0073】

図11に示すように、ISC\_PROGRAM\_SECURITY命令を用いて鍵を読出すことにより確実に攻撃者が非安全モードに戻ることができないように、セキュリティが除去された場合(ISC\_PROGRAM\_SECURITY信号が非安全論理レベルに移る場合)、制御ロジック23aの状態マシンは、一度に1ワードずつ、すべての6個のワードに0を書込むことによりすべての鍵を消去する。これは、ステップ1110で、0をwdata[63:0]バスに与え、ステップ1111でws\_b[0]信号を(論理0値で)アサートし、次にステップ1112~1117で一度に1つずつws\_b[0:0]信号からws\_b[5:0]信号を連続的にストロープし、その後ステップ1118でセキュリティ状態を変更して非安全モードに入り、最後にステップ1119でwdata[6



3:0] 論理0値を解放することによりなされる。したがって、電池でバックアップされたメモリ23を非安全モードにしようすると、鍵レジスタ23bにある値のすべてが消去されてしまう。

#### 【0074】

鍵メモリ23が安全モードにあるかどうかを伝達するために、制御ロジック23aは、鍵メモリ23が安全モードで動作していることを示すために安全モード信号をバス26（単一の線であってもよい）で構成ロジック29に送る。この信号が非安全モードに切換わるならば、構成ロジック29は構成メモリ12から設計をクリアする。なお、鍵が鍵レジスタ23bに記憶されておりかつ鍵メモリ23が安全モードにあるとしても、暗号化されていないビットストリームは構成ロジック29により構成メモリ12にロードされ得る。

#### 【0075】

##### 鍵のロード、複数の暗号化鍵

ユーザが設計の詳細を知ることのできない安全モードにPLDが入る前に、解読鍵はPLDにロードされなければならない。図3に示す実施例では、鍵はJTAGポートを介してロードされる。

#### 【0076】

この発明の特徴として、暗号化鍵はこのJTAGポート20を介してロードされる。JTAGプログラマは基板のテスト中に暗号化鍵をロードすることが求められる。鍵を記憶するためのRAMが非安全モードにあるとき、ユーザはそれへの完全なアクセスを有しており、設計が暗号化されていたとしても鍵および設計の両方を読み出すことができる。鍵をテストしながら鍵を使用できるのでこれは設計者にとって有用である。そして、設計者が動作に満足すると、別の命令をJTAGポートを介して送り、鍵メモリを安全モードにすることができる。鍵メモリが安全モードになってしまうと、鍵は読み出すことができない。さらに、鍵メモリを安全モードから非安全モードに移すことで、メモリ初期化処理を起動する回路を活性化することにより鍵が消去される。（以下に記載する図15は、この機能を行なうための状態マシンを示す。）

この発明のある局面に従えば、設計を暗号化するのに2つ以上の鍵が用いられて

もよい。たとえば、3つの鍵が用いられる場合、まず第1の鍵を用いてビットストリームを暗号化し、第2の鍵を用いて得られた暗号化されたビットストリームを再び暗号化し、最後に第3の鍵を用いて得られた二重に暗号化されたビットストリームを再び暗号化する。この三重に暗号化されたビットストリームは、たとえば、PLDを保持するプリント回路基板上のPROMまたはフラッシュメモリに記憶される。

#### 【0077】

解読については、これらの鍵を連続して（逆の順序で）用いて、暗号化されたビットストリームを繰返し解読する。さらに、特定の設計を解読するのに使用されるよりも多くの鍵がPLDに記憶されている場合、暗号化されたビットストリームは、暗号化されていない部分に、いくつかの鍵が使用されるべきかの指示と、第1の鍵のアドレスとを含み得る。そのような実施例では、攻撃者は一度に1つの鍵に対処するだけでよいので、攻撃者は容易にビットストリームを解読できてしまうかもしれない。代替的に、鍵自体が、それらが最初の鍵か、中間の鍵か、最後の鍵か、または唯一の鍵であるかを示してもよい。したがって、同じPLDを、異なった機能（異なった設計で構成される）を行なうように何回かに分けてプログラムすることができ、異なった鍵の値についての情報を、設計者の1人へのみまたは幾人かに利用可能とすることができる。したがって、どちらの設計も同じPLDに実現される（何回かに分けて）としても第1の設計者が第2の設計について知らないことがあり得る。

#### 【0078】

図3に関し、構成ロジック29は、図1の構成ロジック14を超える追加のロジックを含む。図1の構造におけるように、構成アクセスポート21上のビットストリームは、ワードとして、ある実施例では32ビットのワードとして扱われる。通常、ビットストリームの開始におけるまたはその付近におけるワードのうちの一つかが、ヘッダ情報、たとえばビットストリームの長さ、構成データの開始アドレスを含む。この発明のビットストリームにとって新しいことは、ビットストリームが暗号化されているかどうかについての指示と、ビットストリーム中の構成データを解読するための鍵のアドレスとである。



できるが、攻撃者は鍵に記憶された値を知ることができない。したがって、暗号化されていない設計を讀出したり複製したりすることはできない。このセキュリティを達成するために、いくつかのステップがとられる。

【0087】

#### 安全モードの保存（改ざんの証明）

ある実施例では、PLDの構成ロジック29には2つのセキュリティフラグがある。一方は、解読鍵が安全にされたかどうかを示し、他方は、設計が解読された設計であり保護されなければならないかどうかを示す。JTAGロジック13

（図3）がISC\_\_PROGRAM\_\_SECURITY命令で安全モードを選択する場合、制御ロジック23a（図10a）のsecure\_\_keyフラグがセットされる。PLDにロードされるビットストリームが、そのビットストリームの設計データが暗号化されているという指示を有するならば、構成ロジック29（図示せず）のsecure\_\_designフラグがセットされる。いずれかのフラグが後に設定解除されるならば、構成メモリの全体がクリアされ、解読された設計が除去される。secure\_\_keyフラグが（ISC\_\_PROGRAM\_\_SECURITY命令により）リセットされた場合、鍵も消去される。

【0088】

図15は、設計クリア機能を行なうための状態マシンを示す。secure\_\_designフラグがセットされると、状態マシンは状態S1にはいる。この状態は、secure\_\_designフラグの安全モードから非安全モードへの変化をモニタする。安全設計モードが持続する限り、状態マシンは状態S1に留まる。変化が起きると、状態マシンは状態S2に入り、データを構成メモリ12にシフトするためのデータシフトレジスタがリセットされ、構成メモリビットのためのすべてのデータ線に0が与えられる。次に、状態マシンは、アドレスされたフレームのワード線がアサートされる状態S3に移る。この結果、データシフトレジスタ線上の0が、アドレスされたフレームのメモリビットに書込まれる。質問Q1がアドレスされるべきフレームがさらにあることを示す場合、状態マシンは、フレームアドレスが進められる状態S4に移り、状態マシンは状態S3に戻る。質問Q1がアドレスされるべきフレームがこれ以上ないことを示すとき、処

理は終わり構成メモリがクリアされる。

#### 【0089】

鍵が攻撃者によってアクセスされるのを保護する必要がある。設計を含むシステムが末端の消費者に利用可能となる前に鍵のロードが行なわれる。設計者が設計を開発する処理をしているとき、彼らは、非安全モードでPLDを動作させてデバッグしたいことがある。このデバッグ操作を可能とし、かつ鍵のセキュリティを保護するために、鍵ロード処理は、すべての鍵レジスタをクリアすることにより非安全モードで始まる。鍵がロードされている間および検証のため鍵がリードバックされている間、安全鍵フラグは非安全モードに維持されなければならない。構成ビットストリームがロードされ解読されている間も、安全鍵フラグは非安全モードに維持されなければならない。しかし、安全鍵フラグが一旦セットされると、安全鍵フラグを非安全モードに戻すことにより、すべての鍵をクリアし、また図15の状態マシンの動作を起動する。そのため、鍵がクリアされるのみならず構成もクリアされる。

#### 【0090】

##### リードバック攻撃およびリードバック不能化

いくつかのFPGAは、ビットストリームがFPGAからリードバックされることを可能としており、そのためユーザは設計をデバッグしたりFPGAのフリップフロップから状態マシン情報を獲得し得る。設計がリードバック動作のために再暗号化されないとすれば、ビットストリームをリードバックする動作が、暗号化されていないビットストリームに見えるように暴露してしまうだろう。

#### 【0091】

設計のなお一層のセキュリティは、暗号化された設計がFPGAにロードされるときリードバックを不能化することにより提供される。ある実施例では、解読鍵も安全にされた場合にのみリードバックが不能化される。

#### 【0092】

図16は、構成メモリをロードしリードバックするための構造のブロック図を示す。ある実施例では、構成ロジック29は、以下の2つの条件が存在するときリードバックを防ぐ：(1) データバス26（図3および図10を参照）上のセキ

セキュリティ状態線が鍵が安全モードにあることを示し、かつ(2)構成ロジック29が、ビットストリームが暗号化されていることを示す構成ビットストリーム中の演算コードに応答したときである。両方のキーが安全にされていないかまたはビットストリームが暗号化されていない場合に、リードバックを可能化することができる。他の実施例では、異なった条件が、リードバックを可能化するかどうかを制御する。

#### 【0093】

構成ロジック29が、リードバックが行なわれるべきことを示すヘッダをビットストリーム中に受けるとき、それは、フレームアドレスレジスタに記憶されたフレームアドレスを線107に送り、このフレームアドレスは、バス109のアドレスされた線を選択するためにアドレスデコーダ110によってデコードされる。次に、線108上のワード線可能化信号がアサートされ、これがバス109の選択されたワード線をアサートし、選択されたワード線によりアドレスされるメモリセルがその値をn本のデータ線102に与える(nはフレーム長であり構成ロジック29に記憶される)。そして、構成ロジック29はライン104上のロード信号をアサートして、データのフレームを(並列に)データシフトレジスタ101にロードする。次に、構成ロジック29は、線105上のシフト信号をアサートして、これによりデータシフトレジスタ101は、バス103上の32ビットのワードのデータのフレームをフレームデータ出力レジスタ(図4d)にシフトアウトし、そこから構成アクセスポート21(図3)上の出力ビットストリームにシフトアウトする。

#### 【0094】

解読がビットストリームに示される場合、構成ロジック29は内部フラグをセットしてこれを示す。これらのフラグがセットされ、かつ鍵メモリ23がバス26上のセキュリティ状態信号により示されるように安全モードにある場合、構成ロジック29は、回線108上のワード線可能化信号を不活性に保ち、かつ回線104および105上のロード信号およびシフト信号を不活性に保ってリードバックを防ぐことにより、ビットストリーム中のリードバックコマンドに応答する。しかしながら、鍵メモリ27が安全モードになれば、設計が暗号化されていた

としても、リードバックは許可され、テストおよびデバッグが可能となる。

#### 【0095】

##### 部分的再構成攻撃および防止

いくつかのFPGAは、FPGAの部分的再構成を可能にしたり、別々の開始アドレスおよび別々の書込命令を用いて設計の異なった部分がFPGAの異なった部分にロードされることを可能にする。攻撃者は、設計を部分的に再構成してブロックRAMまたはフリップフロップの内容を出力ポートに直接読出したり、またはセクションを既存の設計に追加して、設計を知るために使用できる情報を読出すことにより、設計を知ろうと企てるかもしれない。たとえば、攻撃者は、PLDを暗号化されていない設計で部分的に再構成するかもしれないが、その唯一の目的は暗号化された設計についての情報を抽出することである。そのようなトロイの木馬設計は、別のビットストリームでPLDにロードされるか、既存の暗号化されたビットストリームに付加されるかもしれない。たとえば、攻撃者が、FPGAのブロックRAMにロードされた状態マシン設計を知ることに興味を持った場合、トロイの木馬設計は、ブロックRAMのアドレスを循環しブロックRAMデータの内容をパッケージピンに送るロジックを含むかもしれない。

#### 【0096】

攻撃者がそのような変更を加えることを防ぐために、当初の設計が暗号化される場合、構成ロジック29は、解読つきの構成が一旦起動されると部分的再構成を禁じる。構成ロジック29は、解読演算コードを有するヘッダが処理されてしまうと後続の書込命令を禁じる。また、構成ロジック29は、暗号化なしの構成が行なわれると解読付きの構成を禁じる。構成ロジック29は、解読命令が受信された後、構成メモリへの書込を行なうヘッダを無視し、設計の暗号化されていない部分がロードされてしまった場合、解読コマンドを有するヘッダを無視することにより、これらの制限を達成する。したがって、いずれかの演算コードが解読付きの書込が用いられていることを示すならば、PLDは単一の書込命令のみを受付ける。

#### 【0097】

##### 追加の実施例

図面の上述の説明はいくつかの実施例についての詳細を与えた。しかしながら、多くの追加の実施例も可能である。たとえば、上述した暗号ブロック連鎖アルゴリズムの代わりに、ブロックサイズよりも小さいユニットで、たとえば一度に1個の8ビットバイトでデータを暗号化することのできる暗号フィードバックモードと呼ばれる暗号化方法を用いることができる。この暗号フィードバックモードは、Schneierにより同書の200から203頁に記載される。

#### 【0098】

さらに別の実施例では、暗号化が用いられる場合、アドレス0で始まってすべてのビットストリームがロードされなければならない。この実施例のある実現化例は、暗号化を特定する演算コードが受信されたとき、開始フレームアドレスレジスタFAR（図6）にロードされたアドレスをアドレス0と置換する。

#### 【0099】

さらに別の実施例では、開始アドレスおよび設計データの両方を暗号化する。この実施例では、暗号化されていない設計データで可能なように、異なったフレームアドレスで始まる暗号化された設計データのいくつかのセグメントをロードすることが可能である。

#### 【0100】

別の実施例では、鍵メモリ23などの鍵メモリに記憶される鍵データは、後に続く鍵の数を明示する。この実施例の変形例では、鍵データは、その鍵に先行する鍵の数も明示する。もし攻撃者が、設計者が意図した最初の鍵アドレスと異なる鍵アドレスを与えるならば、構成が中止されてもよい。さらに、鍵内に明示される数の鍵が使用されてしまうまで、暗号化が行われる。

#### 【0101】

別の実施例では、鍵メモリが非安全モードにあるとき鍵がリードバックされることを可能にする代わりに、鍵は、パリティビットまたはCRCチェックビットを含み、これらのビットのみをリードバックして鍵が正しくロードされたかを確認することができる。この実施例は、ある設計者に知られている鍵を別の設計者に秘密にすることができ、PLDが異なった設計をロードするために複数回に分けて使用されるときに有用である。



## 【0102】

上述のCRCチェックサム計算に関し、CRCチェックサムを、設計が暗号化される前または後のいずれかに計算する実施例が提供され得る。もちろん、設計データが暗号化される前に、ビットストリームに付加されたチェックサムが計算されるならば、対応のチェックサムは、解読された後に設計データに対してPLD内で計算されなければならない。同様に、設計データが暗号化された後に、ビットストリームに付加されたチェックサムが計算されるならば、PLDは、設計データが解読される前に受信ビットストリームに対して対応のチェックサムを計算しなければならない。

## 【0103】

解読鍵をロードする処理に関し、図8に示す処理を用いる場合、デバイスプログラマを用いて解読鍵をロードする必要はない。鍵は、基板検査手続の一部としてロードされればよい。

## 【0104】

2つ以上のPLDをプログラムするために上述の構造および方法を用いることも可能である。いくつかのデバイスをデジタイズチェーン接続に構成しビットストリームをデバイスに直列に通過させるか、またはデバイスを直列にアドレスすることにより、2つ以上のPLDまたはFPGAをプログラムするために単一のビットストリームを用いることは周知である。デバイスのうち1つ以上が暗号化された設計データを受信するものである場合いくつかのPLDをそのような構成に配置することが可能である。

## 【0105】

さらに別の実施例として、暗号化された設計データを有するビットストリームに対して単一のアドレスのみが特定されてもよい実施例が記載されたが、別の実施例では、好ましくは暗号化されたいくつかのアドレスが、設計の別々の部分をロードするために特定されてもよい。さらに、これらの別々の部分は同じ暗号化鍵を用いてもよく、またはこれらの別々の部分は、異なった暗号化鍵および鍵の異なった組を用いてもよい。

## 【0106】

以上の説明から明らかとなる変形はこの発明の範囲に含まれるものとする。

【図面の簡単な説明】

【図1】 先行技術のFPGAにおける機能的関係の図である。

【図2a】 先行技術のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図2b】 先行技術のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図2c】 先行技術のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図2d】 先行技術のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図3】 この発明のある実施例に従うFPGAにおける機能的関係の図である。

【図4a】 この発明のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図4b】 この発明のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図4c】 この発明のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図4d】 この発明のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図5a】 例示の暗号化されていないビットストリームおよび暗号化されたビットストリームの図である。

【図5b】 例示の暗号化されていないビットストリームおよび暗号化されたビットストリームの図である。

【図6】 構成ロジック29ならびに解読器24に至るバス27およびバス28における線の図である。

【図7a】 この発明のある実施例に用いられる三重暗号化での外部暗号ブロック連鎖のための修正された開始値の図である。

【図7b】 図7aで使用される対応の開始値および解読処理の図である。

【図8】ビットストリームを処理するための操作の図である。

【図9】鍵の順序を評価するために解読器24によって実現される状態マシンの図である。

【図10a】図3の鍵メモリ23の構造の図である。

【図10b】図10aのメモリセルの構造の図である。

【図11】非安全とされたときに鍵を消去するために図10aの制御ロジック23aによって行なわれるステップの図である。

【図12】図10aの電池供給スイッチを詳細に示す図である。

【図13】図12の電池供給スイッチのレベルシフト回路および電圧検出回路の図である。

【図14】図12の電池供給スイッチのレベルシフト回路および電圧検出回路の図である。

【図15】安全モードから抜けたとき設計を消去するための状態マシンの図である。

【図16】暗号化が存在するとき不能化される線を含む、構成メモリをロードし構成をリードバックするための要素のブロック図である。

【図1】

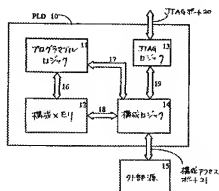
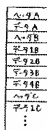


Fig. 1  
PRIOR ART

【図2 a】

Fig. 2a  
PRIOR ART

【図2 b】



Fig. 2b PRIOR ART

【図2 c】

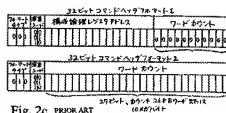


Fig. 2c PRIOR ART

【図2 d】

命令管理レジスタ アドレス	命令管理レジスタ内容
0000	命令管理レジスタ (CR0)
0001	命令管理レジスタ (CR1)
0010	命令管理レジスタ (CR2)
0011	命令管理レジスタ (CR3)
0100	命令管理レジスタ (CR4)
0101	命令管理レジスタ (CR5)
0110	命令管理レジスタ (CR6)
0111	命令管理レジスタ (CR7)
1000	命令管理レジスタ (CR8)
1001	命令管理レジスタ (CR9)
1010	命令管理レジスタ (CR10)
1011	命令管理レジスタ (CR11)
1100	命令管理レジスタ (CR12)
1101	命令管理レジスタ (CR13)
1110	命令管理レジスタ (CR14)
1111	命令管理レジスタ (CR15)

Fig. 2d  
PRIOR ART



【X 4 d】

[illegible]

Fig. 4d

【図 5 a】

[illegible]

Fig. 5a



【図7 a】

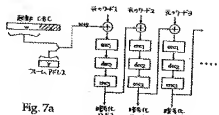


Fig. 7a

【図7 b】

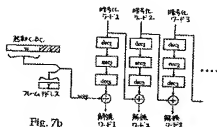


Fig. 7b

【図8】

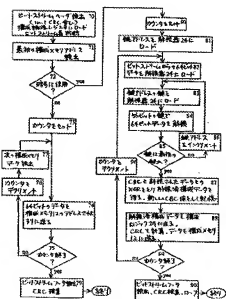


Fig. 8



【図9】

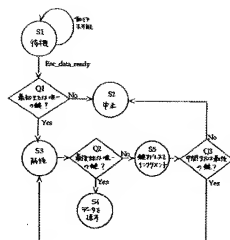


Fig. 9

【図10a】

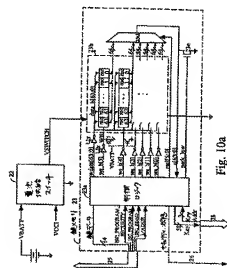
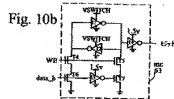
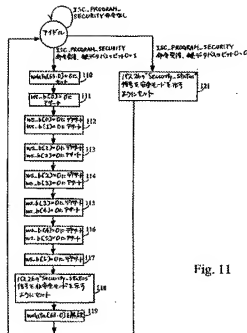


Fig. 10a

【X】 1 0 b



【例 1-1】



【図12】

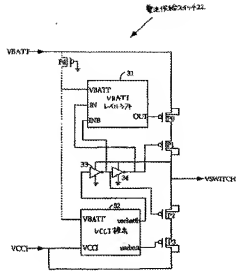


Fig. 12

【図13】

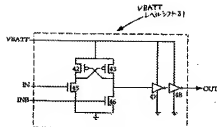


Fig. 13

【図14】

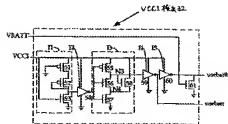


Fig. 14

【図15】

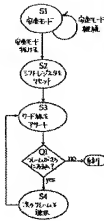


Fig. 15

【図16】

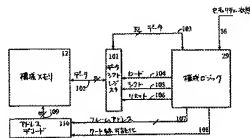


Fig. 16

W/O 02/04/96

PCY 02/04/96

1/16

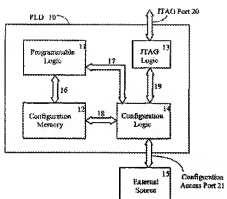


Fig. 1  
PRIOR ART

WI/02/4876

PCT/03/04954

2/16

Header A
Data A
Header B
Data B
Data 2B
Data 3B
Data 4B
Data 5B
Header C
Data 3C
:
:

Fig. 2a  
PRIOR ART

Binstream header Config. Logic Reg. address	Config. Logic Register data contents
0000	Cycle Redundancy Check (CRC)
0001	Frame Address
0010	Frame Data Format
0011	Frame Data Output
0100	Command
0110	Control
0111	Signal
1000	Design Chain Output
1001	Configuration Output
1010	Reserved
1011	Frame Length
1100	Reserved
1101	Reserved
1110	Reserved
1111	Reserved

Fig. 2d  
PRIOR ART

32-bit Command Header Format 1

Format Type	Op- code	Config. Logic Register address	Word count
0010	00	00000000000000000000000000000000	00000000000000000000000000000000

Fig. 2b PRIOR ART

31 bits, example 2048 words

32-bit Command Header Format 1

Format Type	Op- code	Config. Logic Register Address	Word Count
0010	00	00000000000000000000000000000000	00000000000000000000000000000000

32-bit Command Header Format 2

Format Type	Op- code	Word Count
0110	00	00000000000000000000000000000000

27 bits, example 2.68 million words or 10 megabytes

Fig. 2c PRIOR ART

3/18

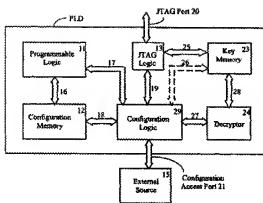


Fig. 3

WD 02/4876

PL3/02/01/02/05

4/16

Header A	
Data A	
Header B	
Data B	
Header C	
Data C	
Header D	
Data D	
Data 1D	
Data 2D	
Data 3D	
Data 4D	
Data 5D	
Data 6D	
Data 7D	
Data 8D	
Data 9D	
Data 10D	
Data 11D	
Data 12D	
Data 13D	
Data 14D	
Data 15D	

Fig. 4a

Electron Header		Config. Logic Register	
Config. Logic Register Address		data contents	
0000	0000	Cycle Refreshment Check (CRC)	
0001	0001	Frame Address	
0010	0010	Frame Data Input	
0011	0011	Frame Data Output	
0100	0100	Control	
0101	0101	Status	
0110	0110	Error Check Output	
0111	0111	Configuration Option	
1000	1000	Reserved	
1001	1001	Frame Length	
1010	1010	Cluster block addressing (CBA) status	
1011	1011	Index key address	
1100	1100	Reserved	
1101	1101	Reserved	
1110	1110	Reserved	
1111	1111	Reserved	

Fig. 4d

32-bit Command Header Format 1

Format Type	Op code	Config. Logic Register address																Word count							
0011	0011	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Fig. 4b

11 bits, counts 2048 words

32-bit Command Header Format 1

Format Type	Op code	Config. Logic Register Address																Word Count							
0011	0011	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0010	0010	0010	0010	0010	0010	0010	0010

32-bit Command Header Format 2

Format Type	Op code	Word Count																									
0110	0110	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Fig. 4c

27 bits, counts 2.68 million words or 10 megabytes



WORD24479

PC145046406

5/16

```

NAME:TYPEC7_VPS2170
Dummy word 1:11111111111111111111111111111111
Byte word 1:01010101010101010101010101010101
Type 1 write 1 words to CPU 001:00000000000000000000000000000000
MPC Lockout 00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
Data word 0 00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
Data word 0 00000000000000000000000000000000
Type 1 write 1 words to ID 001:00000000000000000000000000000000
Data word 0 00000000000000000000000000000000
Type 1 write 1 words to MISC 001:10000000000000000000000000000000
Data word 0 00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
MPCV command 00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
Data word 0 00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
MPCV command 00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
MPCV command 00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
Data word 1 00000000000000000000000000000000
Data word 2 00000000000000000000000000000000
Data word 3 00000000000000000000000000000000
Data word 4 00000000000000000000000000000000
Data word 5 00000000000000000000000000000000
Data word 6 00000000000000000000000000000000
Data word 7 00000000000000000000000000000000
Data word 8 00000000000000000000000000000000
Data word 9 00000000000000000000000000000000
Type 1 NO OP 001:00000000000000000000000000000000
Type 1 NO OP 001:00000000000000000000000000000000
Type 1 NO OP 001:00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
Data word 0 00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
MPCV command 00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
Data word 0 00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
MPCV command 00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
Data word 0 00000000000000000000000000000000
Type 1 write 1 words to CPU 001:10000000000000000000000000000000
MPCV command 00000000000000000000000000000000
Type 1 NO OP 001:00000000000000000000000000000000

```

Fig. 5a

PC TATSUJIMURA

[illegible]

Fig. 5b



W/O 12/04/04

PC PROCESSING

#16

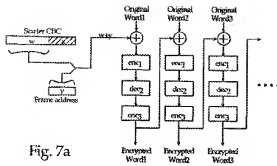


Fig. 7a

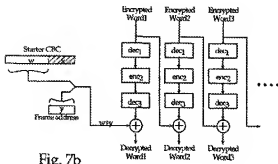


Fig. 7b

WO 2004/05

PC/P0304/2004

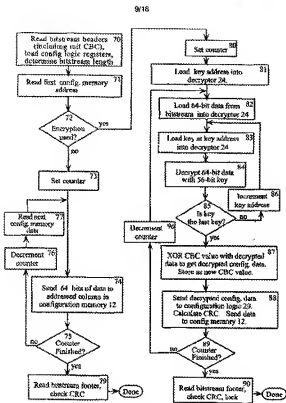


Fig. 8

WTO 02/04/04

PC3/R391/0404

12/16

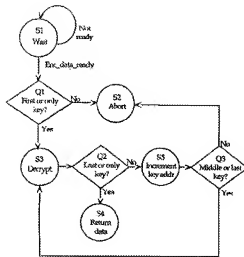
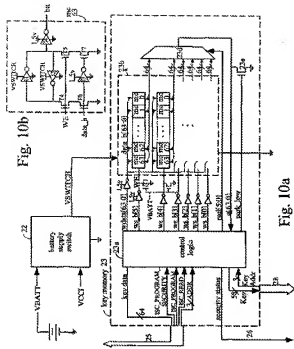


Fig. 9

W11 12/04/05

PC 3/10/05/05/05

11/15



W11 02/44/76

PCT/06/04/96

12/16

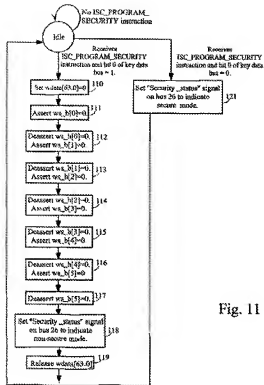


Fig. 11



WO 02/04876

PCT/JP00/0006

13/16

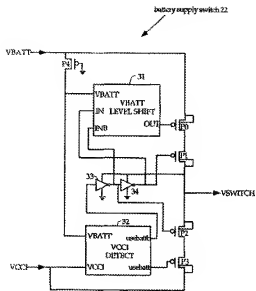


Fig. 12

W3 8224876

PC379946856

14/16

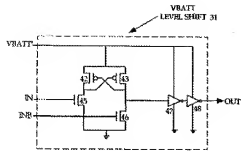


Fig. 13

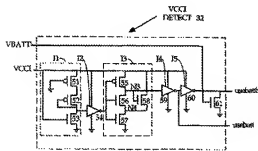


Fig. 14

W/O 0.2/408%

PC 2.0/504/474%

15/16

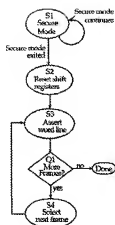


Fig. 15

W012344876

PCT/US2004/000006

16/16

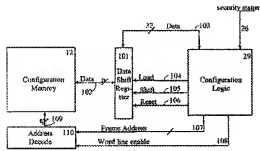


Fig. 16





INTERNATIONAL SEARCH REPORT				International Application No. PCT/JP 01/45058	
Current document classification code	Publication date	Publication number	Publication date	Publication number	Publication date
US 5946142	A	19-10-1999	US	6212639 B1	20-04-2001
US 5615263	A	25-03-1997	NONE		
US 2001015019	A1	23-08-2001	EP	1124339 A2	16-08-2001
			US	2001037452 A1	01-11-2001
			JP	2000301 A	03-07-2001
			US	0146810 A1	28-06-2001
			US	2375410 A	13-11-2002
US 5321704	A	14-06-1994	CA	2003604 A1	17-09-1992
			EP	0495642 A2	22-09-1992
			JP	0003754 A	12-01-1995
			US	5508424 A	06-01-1997

From PCT/JP 01/45058, published under the name of JP 2002-080001 A

フロントページの続き

(74)代理人 100098316

弁理士 野田 久登

(74)代理人 100109162

弁理士 酒井 將行

(72)発明者 バン、レイモンド・シー

アメリカ合衆国、9 5 1 2 0 カリフォルニア州、サン・ノゼ、ファルコン・リッジ・コート、1  
1 3 8

(72)発明者 シー、ウォルター・エヌ

アメリカ合衆国、9 4 0 2 4 カリフォルニア州、ロス・アルトス・ヒルズ、コリー・ドライブ  
、1 9 4 8

(72)発明者 ウォン、ジェニファー

アメリカ合衆国、9 4 5 3 9 カリフォルニア州、フリーモント、エンカント・ウェイ、4 0 5 6  
5

(72)発明者 トリンバーガー、スティーブン・エム

アメリカ合衆国、9 5 1 2 0 カリフォルニア州、サン・ノゼ、シャトー・ドライブ、1 2 6 1

(72)発明者 センティーン、ジョン・エム

アメリカ合衆国、9 4 7 0 9 カリフォルニア州、パークリー、マーティン・ルーサー・キング・  
ジュニア・ウェイ、1 4 3 5、ナンバー・5

(72)発明者 ラオ、カメスワラ・ケイ

アメリカ合衆国、9 5 1 2 9 カリフォルニア州、サン・ノゼ、グレイウッド・ドライブ、1 4 0  
2

F ターム(参考) 5B017 AA03 BA07 CA11

5J042 BA11 CA13 CA20 CA26 DA06

5J104 AA12 AA16 AA32 AA44 AA45 EA04 EA20 NA02 NA27 NA42

PA14



【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】平成17年12月22日(2005.12.22)

【公表番号】特表2004-515001(P2004-515001A)

【公表日】平成16年5月20日(2004.5.20)

【年通号数】公開・登録公報2004-019

【出願番号】特願2002-546976(P2002-546976)

【国際特許分類第7版】

G 0 6 F 12/14

H 0 3 K 19/173

H 0 4 L 9/10

【F I】

G 0 6 F 12/14 3 2 0 B

H 0 3 K 19/173 1 0 1

H 0 4 L 9/00 6 2 1 A

【手続補正書】

【提出日】平成16年11月8日(2004.11.8)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】 構成メモリにより構成される構成可能ロジックと、  
P L D の外部源からビットストリームを受け取るための構造とを含む、  
ビットストリームは、暗号化されていない構成ビットおよび暗号化された構成ビットを  
含む、さらに、

解読鍵を記憶するための鍵メモリと、  
鍵を用いてビットストリーム中の暗号化された構成ビットを解読することにより構成デ  
ータを形成するための解読アルゴリズムを有する解読器と、  
構成データを構成メモリにロードするための構造とを含む、プログラマブルロジックデ  
バイス(P L D)。

【請求項2】 ビットストリームが暗号化されたデータを含むかどうかを示すヘッダ  
情報をビットストリームから読出すための構造と、

ビットストリームが暗号化されたデータを含むことをヘッダ情報が示す場合にはビット  
ストリームを解読器に方向付け、ビットストリームが暗号化されたデータを含まないこと  
をヘッダ情報が示す場合には解読器を飛び越すための構造とをさらに含む、請求項1に記載  
のP L D。

【請求項3】 構成メモリから構成をリードバックするための構造と、  
ビットストリームが暗号化されたデータを含むことをヘッダ情報が示すとき構成をリ  
ードバックするための構造を不能化するための構造とをさらに含む、請求項2に記載のP L  
D。

【請求項4】 P L D が構成された後P L D を再構成するための構造と、  
ビットストリームが暗号化されたデータを含むことをヘッダ情報が示すときP L D を再  
構成するための構造を不能化するための構造とをさらに含む、請求項2に記載のP L D。

【請求項5】 鍵メモリは、複数の解読鍵を記憶するための複数のレジスタを含  
み、  
解読器は、複数の解読鍵のうち少なくとも1つを用いてビットストリーム内のデータ

を解読するための解読アルゴリズムを有する、請求項 1 に記載の P L D。

【請求項 6】 解読器は、複数の解読鍵を記憶するためのレジスタの 1 つから、解読のために別の鍵も使用されるかどうかを示す値を読出す、請求項 5 に記載の P L D。

【請求項 7】 鍵は、当該鍵が、鍵の組のうちの最後の鍵であるか、または最後の鍵でないかを特定する、請求項 5 に記載の P L D。

【請求項 8】 ビットストリーム内の第 1 の群のワードは、第 1 の設計者に既知の第 1 の鍵によって暗号化され、ビットストリーム内の第 2 の群のワードは、第 2 の設計者に既知の第 2 の鍵によって暗号化される、請求項 5 に記載の P L D。

【請求項 9】 鍵メモリを安全モードおよび非安全モードにするための構造をさらに含み、鍵メモリが非安全モードにある間に解読鍵がロードされる、請求項 1 に記載の P L D。

【請求項 10】 安全モードから非安全モードに鍵メモリを移すことにより、解読鍵および構成データの少なくとも 1 つが消去される、請求項 9 に記載の P L D。